A Formally Verified Hybrid System for the Next-Generation Airborne Collision Avoidance System

Jean-Baptiste Jeannin, Khalil Ghorbal, Yanni Kouskoulas, Ryan Gardner, Aurora Schmidt, Erik Zawadzki, André Platzer

A Formally Verified Hybrid System for the Next-Generation Airborne Collision Avoidance System*

Jean-Baptiste Jeannin[1], Khalil Ghorbal[1], Yanni Kouskoulas[2], Ryan Gardner[2], Aurora Schmidt[2], Erik Zawadzki[1], and André Platzer[1]

[1] Carnegie Mellon University
[2] The Johns Hopkins University Applied Physics Laboratory

Abstract. The *Next-Generation Airborne Collision Avoidance System* (ACAS X) is intended to be installed on all large aircraft to give advice to pilots and prevent mid-air collisions with other aircraft. It is currently being developed by the Federal Aviation Administration (FAA). In this paper we determine the geometric configurations under which the advice given by ACAS X is safe under a precise set of assumptions and formally verify these configurations using hybrid systems theorem proving techniques. We conduct an initial examination of the current version of the real ACAS X system and discuss some cases where our safety theorem conflicts with the actual advisory given by that version, demonstrating how formal, hybrid approaches are helping ensure the safety of ACAS X. Our approach is general and could also be used to identify unsafe advice issued by other collision avoidance systems or confirm their safety.

1 Introduction

With growing air traffic, the airspace becomes more crowded, and the risk of airborne collisions between aircraft increases. In the 1970s, after a series of mid-air collisions, the Federal Aviation Administration (FAA) decided to develop an onboard collision avoidance system: the Traffic Alert and Collision Avoidance System (TCAS). This program had great success, and prevented many mid-air collisions over the years. Some accidents still happened; for example, a collision over Überlingen in 2002 occurred due to conflicting orders between TCAS and air traffic control. Airspace management will evolve significantly over the next decade with the introduction of the next-generation air traffic management system; this will create new requirements for collision avoidance. To meet these new requirements, the FAA has decided to develop a new system: the Next-Generation Airborne Collision Avoidance System, known as ACAS X [4,9,13].

Like TCAS, ACAS X avoids collisions by giving vertical guidance to an aircraft's pilot. A typical scenario involves two aircraft: the *ownship* where ACAS X is installed, and another aircraft called the *intruder* that is at risk of colliding with the ownship.

* This research was conducted under the sponsorship of the Federal Aviation Administration Traffic Alert & Collision Avoidance System (TCAS) Program Office (PO) AJM-233 under contract number DTFAWA-11-C-00074. Additionally, support for the basic verification technology used as a foundation for this research was provided by the National Science Foundation under NSF CAREER Award CNS-1054246.

Table 1. Sample advisories and their modeling variables; full table in Technical Report [10]

	ACAS X Specification [12]				Our model	
	Vertical Rate Range		Strength	Delay	Sign	Advisory
Advisory	Min (ft/min)	Max (ft/min)	a_r	d_p (s)	w	$\dot{h}_f$ (ft/min)
DNC	$-\infty$	0	g/4	5	-1	0
MCL	current	$+\infty$	g/4	5	$+1$	current
CL1500	$+1500$	$+\infty$	g/4	5	$+1$	$+1500$
SCL2500	$+2500$	$+\infty$	g/3	3	$+1$	$+2500$
COC	$-\infty$	$+\infty$	Not applicable			

ACAS X is designed to avoid *Near Mid-Air Collisions* (*NMACs*), situations where two aircraft come within $r_p = 500$ ft horizontally and $h_p = 100$ ft vertically [13] of each other. The NMAC definition describes a volume centered around the ownship, shaped like a hockey *puck* of radius r_p and half-height h_p.

In order to be accepted by pilots, and thus operationally suitable, ACAS X needs to strike a balance between giving advice that helps pilots avoid collisions but also minimizes interruptions. These goals oppose each other, and cannot both be perfectly met in the presence of unknown pilot behavior. This paper focuses on precisely characterizing the circumstances in which ACAS X gives advice that is safe. An integral part of the ACAS X development process, this work is intended to help ensure that the design of ACAS X is correct, potentially by identifying ways it should be adjusted.

Airborne Collision Avoidance System ACAS X. In order to prevent an NMAC with other aircraft, ACAS X uses various sensors to determine the position of the ownship, as well as the positions of any intruders [5]. It computes its estimate of the best pilot action by linearly interpolating a precomputed *table* of actions, and, if appropriate, issuing an *advisory* to avoid potential collisions [6] through a visual display and a voice message.

An advisory is a request to the pilot of the ownship to alter or maintain her vertical speed. ACAS X advisories are strictly vertical, and never request any horizontal maneuvering. Table 1 shows a sample of the advisories ACAS X can issue. For example, Do-Not-Climb (DNC) requests that the pilot not climb, and Climb-1500 (CL1500) requests that the pilot climb at more than 1500 ft/min. ACAS X can issue a total of 16 different advisories plus Clear-of-Conflict (COC), which indicates that no action is necessary. To comply with an advisory, the pilot must adjust her vertical rate to fall within the corresponding vertical rate range. Based on previous research [12], the pilot is assumed to do so using a vertical acceleration of strength at least a_r starting after a delay of at most d_p after the advisory has been announced by ACAS X.

At the heart of ACAS X is a table whose domain describes possible configurations for the current state of an encounter, and whose range is a set of scores for each possible action [12,14]. The table is obtained from a Markov Decision Process (MDP) approximating the dynamics of the system in a discretization of the state-space, and optimized using dynamic programming to maximize the expected value of events over all future paths for each action [12]. Near Mid-Air Collision events, for example, are associated with large negative values and issuing an advisory is associated with a small negative value. The policy is to choose the action with the highest expected value from a multilinear interpolation of grid points in this table. ACAS X uses this table, along with some heuristics, to determine the best action to take for the geometry in which it finds itself.

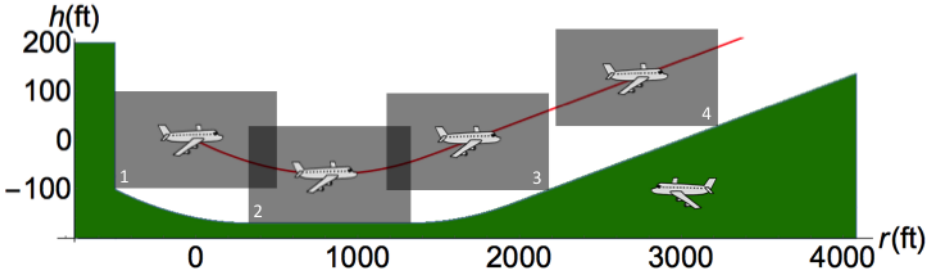

Fig. 1. Trajectory of ownship (red) and safe region for the intruder (green), immediate response

Identifying Formally Verified Safe Regions. Since ACAS X involves both *discrete* advisories to the pilot and *continuous* dynamics of aircraft, it is natural to formally verify it using hybrid systems. However the complexity of ACAS X, which uses at its core a large lookup table—defining 29,212,664 interpolation regions within a 5-dimensional state-space—makes the direct use of hybrid systems verification techniques intractable. Our approach is different. It identifies *safe regions* in the state space of the system where the current positions and velocities of the aircraft ensure that a particular advisory, if followed, prevents all possible NMACs. Then it *compares* these regions to the configurations where the ACAS X table returns this same advisory. Moreover our safe regions are *symbolic* in their parameters, and can thus be easily adapted to new parameters.

Our results provide independent characterizations of the ACAS X behavior to provide a clear and complete picture of its performance. Our method can be used by the ACAS X development team in two ways. It provides a mathematical proof—with respect to a model—that ACAS X is absolutely safe for some configurations of the aircraft. Additionally, when ACAS X is not safe, it is able to identify unsafe or unexpected behaviors and suggests ways of correcting them.

Our approach of formally deriving safe regions then comparing them to the behavior of an industrial system is, as far as we are aware, the first of its kind in the formal verification of hybrid systems. The approach may be valuable for verifying or assessing properties of other systems with similar complexities, or also using large lookup tables, which is a common challenge in practice. Finally, the constraints we identified for safety are fairly general and could be used to analyze other collision avoidance systems.

The paper is organized as follows. After an overview of the method in Sect. 2, we start with a simple two-dimensional model assuming immediate reaction of the pilot in Sect. 3. We extend the model to account for the reaction time of the pilot in Sect. 4, and extend the results to a three-dimensional model in Sect. 5. In Sect. 6, we conduct an initial analysis of ACAS X whereby we compare the advisory recommended by a core component of ACAS X with our safe regions, identifying the circumstances where safety of those ACAS X advisories is guaranteed within our model.

2 Overview of the ACAS X Modelling Approach

To construct a safe region of an advisory for an aircraft, imagine following all allowable trajectories of the ownship relative to the intruder, accounting for every possible position of the ownship and its surrounding puck at every future moment in time. The union of all such positions of the puck describes a potentially unsafe region; for each

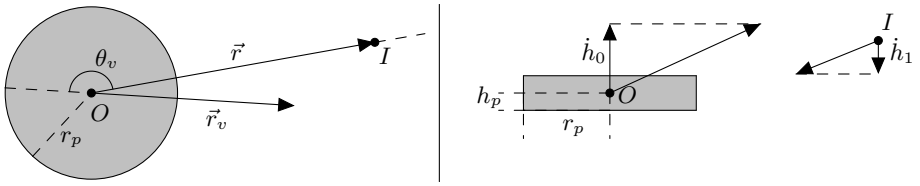

Fig. 2. Top view (left) and side view (right) of an encounter, with NMAC puck in gray

point there exists a trajectory that results in an NMAC. Dually, if the intruder is outside this set, i.e., in the safe region, an NMAC cannot occur in the model.

Fig. 1 depicts an example of a head-on encounter and its associated safe region for the advisory CL1500, projected in a vertical plane with both aircraft. It is plotted in a *frame fixed to the intruder* and centered at the initial position of the ownship. The ownship, surrounded by the puck, starts at position 1 and traces out a trajectory following the red curve. It first accelerates vertically with $g/4$ until reaching the desired vertical velocity of $+1500$ ft/min at position 3. It then climbs at $+1500$ ft/min, respecting the specification of Table 1. The green safe-region indicates starting points in the state space for which the aircraft will remain safe for the duration of the encounter. Note that no safe region exists above the trajectory since the ownship could accelerate vertically at greater than $g/4$ or climb more than $+1500$ ft/min, in accordance with Table 1.

Model of Dynamics. Let us consider an encounter between two planes—ownship O and intruder I, as portrayed in Fig. 2. Following the notation of the ACAS X community [12], let r be the horizontal distance between the aircraft and h the height of the intruder relative to the ownship. We assume that the relative horizontal velocity $\vec{r_v}$ of the intruder with respect to the ownship is constant throughout the encounter. I.e., from a top view, the planes follow straight-line trajectories. Let θ_v be the non-directed angle between $\vec{r_v}$ and the line segment $\vec{r}$. In the vertical dimension, we assume that the ownship's vertical velocity $\dot{h}_0$ can vary at any moment, while the intruder's vertical velocity $\dot{h}_1$ is fixed throughout the encounter. Moreover, we assume that the magnitude of the vertical acceleration of the ownship cannot exceed a_d in absolute value.

For a typical encounter, r varies between $0\,\text{nmi}$ and $7\,\text{nmi}$,[3] h between $-4{,}000\,\text{ft}$ and $4{,}000\,\text{ft}$, r_v between $0\,\text{kts}$ and $1{,}000\,\text{kts}$, and $\dot{h}_0$ and $\dot{h}_1$ between $-5{,}000\,\text{ft/min}$ and $+5{,}000\,\text{ft/min}$. The acceleration a_d is usually $g/2$, where g is Earth's gravitational acceleration. The NMAC *puck* has radius $r_p = 500\,\text{ft}$ and half-height $h_p = 100\,\text{ft}$.

Model of Advisories. Recall that ACAS X prevents NMACs by giving advisories to the ownship's pilot. Every advisory, except COC, has a vertical rate range of the form $(-\infty, \dot{h}_f]$ or $[\dot{h}_f, +\infty)$ for some vertical rate $\dot{h}_f$ (Table 1), which we call the *target vertical velocity*. We model any advisory by its corresponding target vertical velocity $\dot{h}_f$, and a binary variable w for its orientation, whose value is -1 if the vertical rate range of the advisory is $(-\infty, \dot{h}_f]$ and $+1$ if it is $[\dot{h}_f, +\infty)$. This symbolic encoding can represent many advisories and is robust to changes in the ACAS X advisory set.

[3] We use units most common in the aerospace community, even though they are not part of the international system, including nautical miles nmi (1,852 metres), knots kts (nautical miles per hour), feet ft (0.3048 meter) and minutes min (60 seconds).

Following ACAS X design work [12], we assume that the ownship pilot complies with each advisory within d_p seconds, and that she accelerates with acceleration at least a_r to reach the target vertical velocity.

3 Safe Region for an Immediate Pilot Response

We present in this section a simplified version of the dynamics from Sect. 2. We give a hybrid model for this simplified system and prove its safety. The new assumptions will be relaxed in later sections to achieve the safety verification of the full model of Sect. 2.

Model. In this section, we assume that the ownship and intruder are flying head-on ($\theta_v = 180°$). We also assume that the pilot reacts immediately to any advisory ($d_p = 0\,\text{s}$), and that the advisory COC is not allowed. These assumptions will be relaxed in Sect. 4 and Sect. 5. We assume that r is a scalar: if $r \geq 0$ then the ownship is flying towards the intruder, otherwise it is flying away from it. Both cases could require an advisory. Since the ownship and intruder are flying head-on with straight line trajectories, there exists a vertical plane containing both their trajectories. In this plane, the puck becomes a rectangle centered around the ownship, of width $2r_p$ and height $2h_p$, and there is an NMAC if and only if the intruder is in this rectangle (in gray on Fig. 1).

Differential Dynamic Logic and KeYmaera. We model our system using Differential Dynamic Logic d$\mathcal{L}$ [17,18,19], a logic for reasoning about hybrid programs. The logic d$\mathcal{L}$ allows discrete assignments, control structures, and execution of differential equations. It is implemented in the theorem prover KeYmaera [21], that we use to verify our safe regions with respect to our models. All the KeYmaera models and proofs of this paper can be found at `http://www.ls.cs.cmu.edu/pub/acasx.zip`, and statistics in Technical Report [10].

The d$\mathcal{L}$ formula for the model that we use in this section is given in Eq. (1).

$$
\begin{array}{l}
1\ r_p \geq 0 \land h_p > 0 \land r_v \geq 0 \land a_r > 0 \land (w = -1 \lor w = 1) \land C_{\mathsf{impl}}(r, h, \dot{h}_0) \rightarrow \\
2\ [(\ (\ ?\mathsf{true} \cup \dot{h}_f := *; (w := -1 \cup w := 1); ?C_{\mathsf{impl}}(r, h, \dot{h}_0); \mathsf{advisory} := (w, \dot{h}_f)\ ); \\
3\quad a := *; \{r' = -r_v, h' = -\dot{h}_0, \dot{h}_0' = a \,\&\, w\dot{h}_0 \geq w\dot{h}_f \lor wa \geq a_r\} \\
4\ )^*]\ (|r| > r_p \lor |h| > h_p)
\end{array}
\tag{1}
$$

This formula of the form $p \rightarrow [\alpha]q$ says all executions of program α starting in a state satisfying logical formula p end up in a state satisfying q. It is akin to the Hoare triple $\{p\}\alpha\{q\}$ with precondition p and postcondition q. The precondition in Eq. (1) imposes constraints on several constants, as well as the formula $C_{\mathsf{impl}}(r, h, \dot{h}_0)$ (defined below) that forces the intruder to be in a safe region for an initial advisory $(w, \dot{h}_f)$. We cannot guarantee safety if the intruder starts initially in an unsafe region. The postcondition encodes absence of NMAC. Line 2 expresses the action of the ACAS X system. The nondeterministic choice operator $\cup$ expresses that the system can either continue with the same advisory by doing nothing—just testing ?true—this ensures it always has a valid choice and cannot get stuck. Otherwise it can choose a new $\mathsf{advisory}$ $(w, \dot{h}_f)$ that passes the safety condition $C_{\mathsf{impl}}(r, h, \dot{h}_0)$—$\mathsf{advisory}$ will be the next message to the pilot. Line 3 expresses the action of the ownship, first nondeterministically choosing an

arbitrary acceleration ($a := *$) then following the continuous dynamics. The evolution of the variables r, h and $\dot{h}_0$ is expressed by a differential equation, and requires (using the operator $\&$) that the ownship evolves towards its target vertical velocity $\dot{h}_f$ at acceleration a_r (condition $wa \geq a_r$), unless it has already reached vertical velocity $\dot{h}_f$ (condition $w\dot{h}_0 \geq w\dot{h}_f$). Finally, the star * on line 4 indicates that the program can be repeated any number of times, allowing the system to go through several advisories.

Implicit Formulation of the Safe Region. As explained in Sect. 2, we use a frame fixed to the intruder and with its origin at the initial position of the ownship (see Fig. 1).

First case: if $w = +1$ and $\dot{h}_f \geq \dot{h}_0$. Fig. 1 shows, in red, a possible trajectory of an ownship following exactly the requirements of ACAS X. This *nominal* trajectory of the ownship is denoted by $\mathcal{N}$. The pilot reacts immediately, and the ownship starts accelerating vertically with acceleration a_r until reaching the target vertical velocity $\dot{h}_f$—describing a parabola—then climbs at vertical velocity $\dot{h}_f$ along a straight line. Horizontally, the relative velocity r_v remains constant. Integrating the differential equations in Eq. (1) line 3, the ownship position (r_t, h_t) at time t along $\mathcal{N}$ is given by:

$$(r_t, h_t) = \begin{cases} \left(r_v t\,,\ \frac{a_r}{2}t^2 + \dot{h}_0 t\right) & \text{if } 0 \leq t < \frac{\dot{h}_f - \dot{h}_0}{a_r} \qquad (a) \\ \left(r_v t\,,\ \dot{h}_f t - \frac{(\dot{h}_f - \dot{h}_0)^2}{2a_r}\right) & \text{if } \frac{\dot{h}_f - \dot{h}_0}{a_r} \leq t \qquad (b) \end{cases} \tag{2}$$

Recall that in the ACAS X specification, the ownship moves vertically with acceleration of *at least* a_r, then continues with vertical velocity of *at least* $\dot{h}_f$. Therefore all possible future positions of the ownship are *above* the red nominal trajectory. An intruder is safe if its position is always either to the side of or under any puck centered on a point in $\mathcal{N}$, that is:

$$\forall t. \forall r_t. \forall h_t. \big((r_t, h_t) \in \mathcal{N} \rightarrow |r - r_t| > r_p \vee h - h_t < -h_p\big) \tag{3}$$

We call this formulation the *implicit formulation of the safe region*. It does not give explicit equations for the safe region border, but expresses them instead implicitly with respect to the nominal trajectory.

Generalization. The reasoning above is generalized to the case where $\dot{h}_f < \dot{h}_0$, and symmetrically to the case $w = -1$. The most general implicit formulation of the safe region is C_{impl} in Fig. 3, and verified to be safe in KeYmaera:

Theorem 1 (Correctness of implicit safe regions). *The d$\mathcal{L}$ formula given in Eq.* (1) *is valid. That is as long as the advisories obey formula C_{impl} there will be no NMAC.*

Explicit Formulation of the Safe Region. The implicit formulation of the safe region gives an intuitive understanding of where it is safe for the intruder to be. However, because it still contains quantifiers, its use comes at the extra cost of eliminating the quantifiers. An efficient comparison with the ACAS X table, as described in Sect. 6, can only be achieved with a quantifier-free, *explicit formulation*, that we present in this section. We show that both formulations are equivalent. As for the implicit formulation, we derive the equations for one representative case before generalizing them.

Implicit formulation

$$A(t,h_t,\dot{h}_0) \equiv \left(0 \le t < \frac{\max(0,w(\dot{h}_f-\dot{h}_0))}{a_r} \wedge h_t = \frac{wa_r}{2}t^2 + \dot{h}_0 t\right) \vee \left(t \ge \frac{\max(0,w(\dot{h}_f-\dot{h}_0))}{a_r} \wedge h_t = \dot{h}_f t - \frac{w\max(0,w(\dot{h}_f-\dot{h}_0))^2}{2a_r}\right)$$

$$C_{\mathsf{impl}}(r,h,\dot{h}_0) \equiv \forall t.\forall r_t.\forall h_t.\left(r_t = r_v t \wedge A(t,h_t,\dot{h}_0) \rightarrow (|r-r_t| > r_p \vee w(h-h_t) < -h_p)\right)$$

Explicit formulation

$$\mathsf{case}_1(r,\dot{h}_0) \equiv -r_p \le r < -r_p - \frac{r_v \min(0,w\dot{h}_0)}{a_r}$$

$$\mathsf{bound}_1(r,h,\dot{h}_0) \equiv wr_v^2 h < \frac{a_r}{2}(r+r_p)^2 + wr_v\dot{h}_0(r+r_p) - r_v^2 h_p$$

$$\mathsf{case}_2(r,\dot{h}_0) \equiv -r_p - \frac{r_v \min(0,w\dot{h}_0)}{a_r} \le r \le r_p - \frac{r_v \min(0,w\dot{h}_0)}{a_r}$$

$$\mathsf{bound}_2(r,h,\dot{h}_0) \equiv wh < -\frac{\min(0,w\dot{h}_0)^2}{2a_r} - h_p$$

$$\mathsf{case}_3(r,\dot{h}_0) \equiv r_p - \frac{r_v \min(0,w\dot{h}_0)}{a_r} < r \le r_p + \frac{r_v \max(0,w(\dot{h}_f-\dot{h}_0))}{a_r}$$

$$\mathsf{bound}_3(r,h,\dot{h}_0) \equiv wr_v^2 h < \frac{a_r}{2}(r-r_p)^2 + wr_v\dot{h}_0(r-r_p) - r_v^2 h_p$$

$$\mathsf{case}_4(r,\dot{h}_0) \equiv r_p + \frac{r_v \max(0,w(\dot{h}_f-\dot{h}_0))}{a_r} < r$$

$$\mathsf{bound}_4(r,h,\dot{h}_0) \equiv (r_v = 0) \vee \left(wr_v h < w\dot{h}_f(r-r_p) - \frac{r_v \max(0,w(\dot{h}_f-\dot{h}_0))^2}{2a_r} - r_v h_p\right)$$

$$\mathsf{case}_5(r,\dot{h}_0) \equiv -r_p \le r < -r_p + \frac{r_v \max(0,w(\dot{h}_f-\dot{h}_0))}{a_r}$$

$$\mathsf{bound}_5(r,h,\dot{h}_0) \equiv wr_v^2 h < \frac{a_r}{2}(r+r_p)^2 + wr_v\dot{h}_0(r+r_p) - r_v^2 h_p$$

$$\mathsf{case}_6(r,\dot{h}_0) \equiv -r_p + \frac{r_v \max(0,w(\dot{h}_f-\dot{h}_0))}{a_r} \le r$$

$$\mathsf{bound}_6(r,h,\dot{h}_0) \equiv (r_v = 0 \wedge r > r_p) \vee \left(wr_v h < w\dot{h}_f(r+r_p) - \frac{r_v \max(0,w(\dot{h}_f-\dot{h}_0))^2}{2a_r} - r_v h_p\right)$$

$$C_{\mathsf{expl}}(r,h,\dot{h}_0) \equiv \left(w\dot{h}_f \ge 0 \rightarrow \bigwedge_{i=1}^{4}(\mathsf{case}_i(r,\dot{h}_0) \rightarrow \mathsf{bound}_i(r,h,\dot{h}_0))\right) \wedge \left(w\dot{h}_f < 0 \rightarrow \bigwedge_{i=5}^{6}(\mathsf{case}_i(r,\dot{h}_0) \rightarrow \mathsf{bound}_i(r,h,\dot{h}_0))\right)$$

Fig. 3. Implicit and explicit formulations of the safe region for an immediate response

First case: if $w = +1$*,* $r_v > 0$*,* $\dot{h}_0 < 0$ *and* $\dot{h}_f \geq 0$*.* We are in the case shown in Fig. 1 and described in detail above. The nominal trajectory $\mathcal{N}$ is given by Eq. (2)(a) and Eq. (2)(b). The boundary of the (green) safe region in Fig. 1 is drawn by either the bottom left hand corner, the bottom side or the bottom right hand corner of the puck. This boundary can be characterized by a set of equations:

0. positions left of the puck's initial position ($r < -r_p$) are in the safe region;
1. then the boundary follows the bottom left hand corner of the puck as it is going down the parabola of Eq. (2)(a); therefore for $-r_p \leq r < -r_p - \frac{r_v \dot{h}_0}{a_r}$, the position (r, h) is safe if and only if $h < \frac{a_r}{2r_v^2}(r + r_p)^2 + \frac{\dot{h}_0}{r_v}(r + r_p) - h_p$;
2. following this, the boundary is along the bottom side of the puck as it is at the bottom of the parabola of Eq. (2)(a); therefore for $-r_p - \frac{r_v \dot{h}_0}{a} \leq r \leq r_p - \frac{r_v \dot{h}_0}{a_r}$, the position (r, h) is in the safe region if and only if $h < -\frac{\dot{h}_0^2}{2a_r} - h_p$;
3. then the boundary follows the bottom right hand corner of the puck as it is going up the parabola of Eq. (2)(a); therefore for $r_p - \frac{r_v \dot{h}_0}{a_r} < r \leq r_p + \frac{r_v(\dot{h}_f - \dot{h}_0)}{a_r}$, the position (r, h) is safe if and only if $h < \frac{a_r}{2r_v^2}(r - r_p)^2 + \frac{\dot{h}_0}{r_v}(r - r_p) - h_p$;
4. finally the boundary follows the bottom right hand corner of the puck as it is going up the straight line of Eq. (2)(b); therefore for $r_p + \frac{r_v(\dot{h}_f - \dot{h}_0)}{a_r} < r$, the position (r, h) is in the safe region if and only if $h < \frac{\dot{h}_f}{r_v}(r - r_p) - \frac{(\dot{h}_f - \dot{h}_0)^2}{2a_r} - h_p$.

Generalization. The general case is given in the formula C_{expl} of Fig. 3. The cases 1-4 and their associated bounds are for the case $w\dot{h}_f \geq 0$, whereas cases 5 and 6 and associated bounds are for $w\dot{h}_f < 0$. We again use KeYmaera to formally prove that this explicit safe region formulation is equivalent to its implicit counterpart.

Lemma 1 (Correctness of explicit safe regions). *If* $w = \pm 1$*,* $r_p \geq 0$*,* $h_p > 0$*,* $r_v \geq 0$ *and* $a_r > 0$*, then the conditions* $C_{\mathsf{impl}}(r, h, \dot{h}_0)$ *and* $C_{\mathsf{expl}}(r, h, \dot{h}_0)$ *are equivalent.*

4 Safe Region for a Delayed Pilot Response

We generalize the model of Sect. 3 to account for a non-deterministic, non-zero pilot delay, and for periods of time where the system does not issue an advisory (i.e., COC).

Model. In this section, we still assume that the ownship and intruder are flying head-on ($\theta_v = 180°$). We use the same conventions as in Sect. 3 for r and r_v. The model includes an initial period where there is no compliance with any advisory—the ownship accelerates non-deterministically (within limits) in the vertical direction. As before, we derive the safe regions by considering all possible positions of the ownship's puck in all possible trajectories that might evolve in the encounter. To represent pilot delay for an advisory, the model assumes an immediate advisory, and period of non-compliance d_p, representing the time it takes the pilot to respond. To represent COC, the model looks for a safe advisory it can issue d_ℓ in the future if necessary, (d_ℓ being the system delay, and shortest COC) so the period of non-compliance is $d_p + d_\ell$.

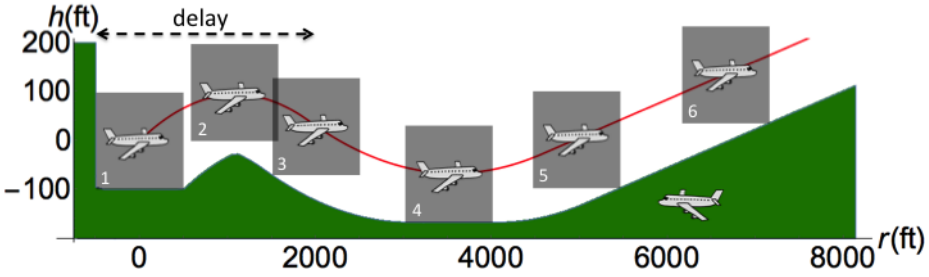

Fig. 4. Trajectory of the ownship (red) and safe region for the intruder (green), delayed response

$$
\begin{array}{l}
1\ r_p \geq 0 \wedge h_p > 0 \wedge r_v \geq 0 \wedge \boldsymbol{a_r > 0 \wedge a_d \geq \wedge d_p \geq 0 \wedge d_\ell \geq 0} \\
2\ \wedge (w = -1 \vee w = 1) \wedge \boldsymbol{D_{\mathsf{impl}}(r, h, \dot{h}_0, d)} \rightarrow \\
3\ [(\ (?\mathsf{true} \cup \dot{h}_f := *; (w := -1 \cup w := 1); \\
4\qquad\qquad \boldsymbol{(d := d_p; ?D_{\mathsf{impl}}(r, h, \dot{h}_0, d)}; \mathsf{advisory} := (w, \dot{h}_f) \cup \\
5\qquad\qquad \boldsymbol{d := d_p + d_\ell; ?D_{\mathsf{impl}}(r, h, \dot{h}_0, d); \mathsf{advisory} := \mathsf{COC}})\ ); \\
6\quad a := *; \boldsymbol{?(wa \geq -a_d); t_\ell := 0;} \\
7\quad \{r' = -r_v, h' = -\dot{h}_0, \dot{h}_0' = a, \boldsymbol{d' = -1, t_\ell' = 1}\ \& \\
8\qquad \boldsymbol{(t_\ell \leq d_\ell)} \wedge (\boldsymbol{d \leq 0} \rightarrow w\dot{h}_0 \geq w\dot{h}_f \vee wa \geq a_r)\} \\
9\ )^*]\ (|r| > r_p \vee |h| > h_p)
\end{array}
\tag{4}
$$

We modify the model of Eq. (1) to capture these new ideas, and obtain the model of Eq. (4), highlighting the differences in **bold**. The structure, precondition (lines 1 and 2) and postcondition (line 9) are similar. The clock d, if positive, represents the amount of time until the ownship pilot must respond to the current advisory to remain safe. Lines 3 to 5 represent the actions of the ACAS X system. As before, the system can continue with the same advisory (?true). Otherwise it can select a safe advisory $(w, \dot{h}_f)$ to be applied after at most delay d_p; or it can safely remain silent, displaying COC, if it knows an advisory $(w, \dot{h}_f)$ that is safe if applied after delay $d_p + d_\ell$. In line 6, the pilot non-deterministically chooses an acceleration ($a := *$), within some limit ($wa \geq -a_d$). The set of differential equations in line 7 describes the system's dynamics, and the conditions in line 8 use the clock t_ℓ to ensure that continuous time does not evolve longer than system delay d_ℓ without a system response ($t_\ell \leq d_\ell$). Those conditions also ensure that when $d \leq 0$ the pilot starts complying with the advisory. The model is structured so that the pilot can safely delay responding to an advisory for up to d_p, and to an advisory associated with COC for up to $d_p + d_\ell$—considering upper bounds on the reaction delay is necessary to get a formal proof of safety. Because of the loop in our model (line 9), the safety guarantees of this theorem apply to encounters whose advisories change as the encounter evolves, encounters with periods of no advisory, and encounters where the pilot exhibits some non-deterministic behavior.

In the rest of the section we use the same approach as in Sect. 3: we first derive an implicit formulation, then an equivalent explicit formulation of the safe region, and prove that the safe region guarantees that the intruder cannot cause an NMAC.

Formulations of the Safe Region. As in Sect. 3, let us place ourselves in the referential centered on the current position of the ownship and where the intruder is fixed, and let us first assume that the ownship receives an advisory $(w, \dot{h}_f)$ such that $w = +1$, and that

Implicit formulation

$$B(t, h_t, \dot{h}_0, d) \equiv 0 \le t < \max(0,d) \land h_t = -\frac{wa_d}{2}t^2 + \dot{h}_0 t$$

$$\mathsf{const} \equiv h_d = -\frac{wa_d}{2}\max(0,d)^2 + \dot{h}_0 \max(0,d) \land \dot{h}_d - \dot{h}_0 = -wa_d \max(0,d)$$

$$D_{\mathsf{impl}}(r, h, \dot{h}_0, d) \equiv \forall t.\forall r_t.\forall h_t.\forall h_d.\forall \dot{h}_d.$$
$$\Big(r_t = r_v t \land (B(t, h_t, \dot{h}_0, d) \lor \mathsf{const} \land A(t - \max(0,d), h_t - h_d, \dot{h}_d))$$
$$\to (|r - r_t| > r_p \lor w(h - h_t) < -h_p)\Big)$$

Explicit formulation

$$r_d = r_v \max(0,d) \qquad \dot{h}_d = \dot{h}_0 - wa_d \max(0,d)$$
$$h_d = -\frac{wa_d}{2}\max(0,d)^2 + \dot{h}_0 \max(0,d)$$
$$\mathsf{case}_7(r) \equiv -r_p \le r \le r_p \qquad \mathsf{bound}_7(r,h) \equiv wh < -h_p$$
$$\mathsf{case}_8(r) \equiv r_p < r \le r_d + r_p \qquad \mathsf{case}_9(r) \equiv -r_p \le r < r_d - r_p$$
$$\mathsf{bound}_8(r,h) \equiv wr_v^2 h < -\frac{a_d}{2}(r - r_p)^2 + wr_v\dot{h}_0(r - r_p) - r_v^2 h_p$$
$$\mathsf{bound}_9(r,h) \equiv wr_v^2 h < -\frac{a_d}{2}(r + r_p)^2 + wr_v\dot{h}_0(r + r_p) - r_v^2 h_p$$
$$D_{\mathsf{expl}}(r, h, \dot{h}_0, d) \equiv \left(\bigwedge_{i=7}^{9}(\mathsf{case}_i(r) \to \mathsf{bound}_i(r,h))\right) \land C_{\mathsf{expl}}(r - r_d, h - h_d, \dot{h}_d)$$

Fig. 5. Implicit and explicit formulations of the safe region for a delayed response

$d \ge 0$. Let us focus on the period of time before the pilot reacts, which we henceforth call delay. During the delay, the ownship can take any vertical acceleration less than a_d in absolute value, therefore its nominal trajectory $\mathcal{N}_d$ is to accelerate the opposite way of the advisory, at acceleration $-a_d$. Horizontally, its speed is constant at r_v. It thus describes a *delay parabola*, in red on Fig. 4, and its position (r_t, h_t) along the nominal trajectory for $0 \le t < d$ is given by $(r_t, h_t) = \left(r_v t, -\frac{a_d}{2}t^2 + \dot{h}_0 t\right)$.

After the delay, i.e., after time d, the nominal trajectory $\mathcal{N}_d$ is the same as a nominal trajectory $\mathcal{N}$ from Sect. 3, translated by time d and by its position at time d given by $r_d = r_t(d)$ and $h_d = h_t(d)$, and starting with vertical velocity $\dot{h}_d = \dot{h}_0 - a_d d$. As in Sect. 3, we can now express the implicit formulation of the safe region:

$$\forall t.\forall r_t.\forall h_t.\big((r_t, h_t) \in \mathcal{N}_d \to |r - r_t| > r_p \lor h - h_t < -h_p\big)$$

Symmetrically, the reasoning of this section extends to the case where $w = -1$. Moreover, we can handle cases where $d < 0$, i.e., after the pilot has reacted, by replacing d by $\max(0,d)$. The generalized implicit formulation of the safe region is given as D_{impl} in Fig. 5. Note that it involves the expression $A(t - \max(0,d), h_t - h_d, \dot{h}_d)$ from Fig. 3 capturing the implicit safe region of Sect. 3 translated by time $\max(0,d)$, vertical height h_d, and starting at vertical speed $\dot{h}_d$. It is proved correct in KeYmaera.

Theorem 2 (Correctness of delayed safe regions). *The $d\mathcal{L}$ formula given in Eq.* (4) *is valid. That is as long as the advisories obey formula D_{impl} there will be no NMAC.*

Similarly as in Sect. 4, we determine an explicit formulation of the safe region, called D_{expl} in Fig. 5 based on Fig. 3, and prove it correct in KeYmaera.

Lemma 2 (Correctness of delayed explicit safe regions). *If $w = -1$ or $w = +1$, $r_p \geq 0$, $h_p > 0$, $r_v \geq 0$, $a_r > 0$, $a_d \geq 0$, $d_p \geq 0$ and $d_\ell \geq 0$ then the two conditions $D_{\mathsf{impl}}(r, h, \dot{h}_0, d)$ and $D_{\mathsf{expl}}(r, h, \dot{h}_0, d)$ are equivalent.*

5 Reduction from 3D Dynamics to 2D Dynamics

In this section, we show that, with respect to our assumptions, any 3-dimensional encounter (Sect. 2) can be reduced to a 2-dimensional encounter (Sect. 3) without loss of generality. This is done using a change of reference frame and a dimension reduction.

For the sake of clarity, let us use a reference frame $(O, \vec{i}, \vec{j}, \vec{k})$ fixed to the ownship (O). In this reference frame, the position of an intruder I is represented by the tuple (x, y, h), and the differential equation system that governs its motion is given by $\dot{x} = r_x$, $\dot{y} = r_y$, $\ddot{h} = a$, where r_x, r_y and a remain constant as time evolves. Therefore, the motion of the encounter can be decoupled into a 2-dimensional horizontal encounter in the reference frame $(O, \vec{i}, \vec{j})$ (horizontal plane) and a 1-dimensional vertical encounter in the reference frame $(O, \vec{k})$. In what follows, we reduce the horizontal encounter from a 2-dimensional motion to a 1-dimensional motion, thereby simplifying the problem conceptually and computationally by reducing its number of variables.

Fig. 6 depicts a top view of a generic encounter. We denote by $\vec{r}$ the position, and $\vec{r}_v$ the velocity, of the intruder relative to the ownship, and by $r_v \geq 0$ the norm of $\vec{r}_v$.

First suppose $r_v > 0$. The idea is to choose a reference frame $(O', \vec{i}', \vec{j}')$ in which one axis $\vec{i}'$ is aligned with $\vec{r}_v$, such that no relative motion happens in the other direction $\vec{j}'$. Its fixed center O' is defined as the orthogonal projection of point O on the direction of $\vec{r}_v$. The unit vector $\vec{i}'$ is defined as $\frac{\vec{r}_v}{r_v}$, and $\vec{j}'$ is a unit such that $(O', \vec{i}', \vec{j}')$ is positively oriented.

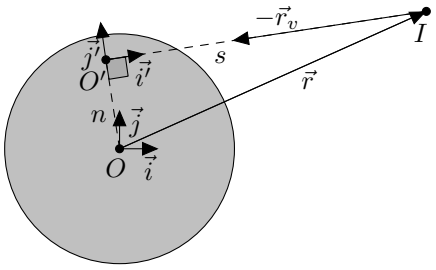

Fig. 6. Top view of the two reference frames

Let $\vec{v}_{|O}$ (resp. $\vec{v}_{|O'}$) denote the coordinates of a vector $\vec{v}$ relative to the reference frame $(O, \vec{i}, \vec{j})$ (resp. $(O', \vec{i}', \vec{j}')$). Then, the coordinates for $\vec{r}$ and $\vec{r}_v$ are: $\vec{r}_{|O} = (x, y)$, $\vec{r}_{v|O} = (r_x, r_y)$, $\vec{r}_{|O'} = (s, n)$ and $\vec{r}_{v|O'} = (-r_v, 0)$. The scalar product $\vec{r} \cdot \vec{r}_v$ and the cross product $\vec{r} \times \vec{r}_v$ are independent of the horizontal reference frame, therefore:

$$xr_x + yr_y = -sr_v \qquad xr_y - yr_x = nr_v \tag{5}$$

Given r_x and r_y, Eqns. (5) imply that the coordinates (x, y) are uniquely determined by the choice of (s, n), as long as $r_v \neq 0$ (using $r_v^2 = r_x^2 + r_y^2$). For any 2-dimensional configuration, the encounter can thus be considered a head-on encounter where s plays the role of r and where a new puck radius, denoted s_p, plays the role of r_p.

Let us now determine the radius s_p of the dimension-reduced encounter, and prove that the absence of NMAC in $(O, \vec{i}, \vec{j})$—characterized by $r^2 > r_p^2$—is equivalent to the absence of NMAC in $(O', \vec{i}', \vec{j}')$—characterized by $s^2 > s_p^2$. Using (5):

$$r_v^2 r^2 = r_v^2(x^2 + y^2) = (xr_x + yr_y)^2 + (xr_y - yr_x)^2 = r_v^2(s^2 + n^2) \ .$$

Table 2. Summary of the points of the state space at which we examined ACAS X

	Range r (ft)	Relative speed r_v (ft/s)	Angle θ_v (degrees)	Relative altitude h (ft)	Vertical rates $\dot{h}_0, \dot{h}_1$ (ft/s)	Previous advisory
Min value	1,500	100	180°	-4,000	-41.67	None
Max value	200,000	2,200	180°	4,000	41.67	None
Number of values	80	10	1	33	13^2	1

Since $r_v \neq 0$, this implies $r^2 = s^2 + n^2$. Therefore, $r^2 > r_p^2$ if and only if $s^2 + n^2 > r_p^2$ or equivalently $s^2 > r_p^2 - n^2$. If $r_p^2 - n^2 < 0$, the direction of the vector $\vec{r}_v$ does not intersect the puck, the inequality $s^2 > r_p^2 - n^2$ is trivially true, and the encounter is safe. If $r_p^2 - n^2 \geq 0$, we choose the new puck radius s_p for the dimension-reduced encounter as $s_p = \sqrt{{r_p}^2 - n^2} \geq 0$, and the safety condition in $(O', \vec{i}', \vec{j}')$ becomes $s^2 \geq s_p^2$. When $\theta_v = 180^\circ$, one has $s = r$, $n = 0$ and $s_p = r_p$ as in Sect. 3–4.

As the encounter evolves in $(O, \vec{i}, \vec{j})$ along $\dot{x} = r_x, \dot{y} = r_y$, its dimension-reduced version evolves in $(O', \vec{i}', \vec{j}')$ along the differential equations $\dot{s} = -r_v, \dot{n} = 0$, obtained by differentiating Eqns. (5) and canceling r_v. The following proposition, proved in KeYmaera, combines both dynamics and shows that the absence of an NMAC of radius r_p in $(O, \vec{i}, \vec{j})$ is equivalent to the absence of an NMAC of radius s_p in $(O', \vec{i}', \vec{j}')$.

Proposition 1 (Horizontal Reduction). *The following dℒ formula is valid*

$$\begin{aligned}\big(xr_x + yr_y = -sr_v \wedge xr_y - yr_x = nr_v \wedge x^2 + y^2 = n^2 + s^2 \wedge r_v^2 = r_x^2 + r_y^2\big)\\ \rightarrow [\dot{x} = r_x, \dot{y} = r_y, \dot{s} = -r_v, \dot{n} = 0]\left(x^2 + y^2 > r_p^2 \leftrightarrow s^2 > r_p^2 - n^2\right)\end{aligned} \quad (6)$$

Observe that the horizontal NMAC condition in $(O', \vec{i}', \vec{j}')$ only depends on the change of one variable rather than two. The proposition also applies to the special case $r_v = 0$. In this case the origin O' is no longer defined, and Eqns. (5) are trivially true. The variables s and n are constants ($\dot{s} = 0, \dot{n} = 0$), their initial values are only restricted by the condition $n^2 + s^2 = x^2 + y^2$ in the assumption of the proposition, but they are not unique. When the relative position between the two aircraft does not evolve over time, if the intruder is at a safe distance initially, the encounter is still safe for all time.

6 Initial Examination of the Safety of ACAS X

In this section, we use Theorem 1 to check the safety of advisories given by ACAS X. We focus on Run 12 (July 2014) of the optimized logic tables, a core component of ACAS X. The full policy of the system is built on these lookup tables and incorporates additional components to handle various operational scenarios. We compare the ACAS X table to the explicit regions where the pilot reacts immediately (Sect. 3). For a given initial state of an encounter, we query the *first* advisory issued by ACAS X and check its safety as identified in Theorem 1. In a real scenario, ACAS X could later strengthen or reverse the first advisory as the encounter evolves. But the safety of the first advisory is critical from an operational prospective as later changes are undesirable.

Our initial analysis considers a nominal set of discrete states—summarized in Table 2—of the ACAS X MDP model where no advisory has yet been issued. All compared states are head-on encounters: in a sense, they are the most obviously dangerous configurations. For those states, the ACAS X advisories are compared against the safe

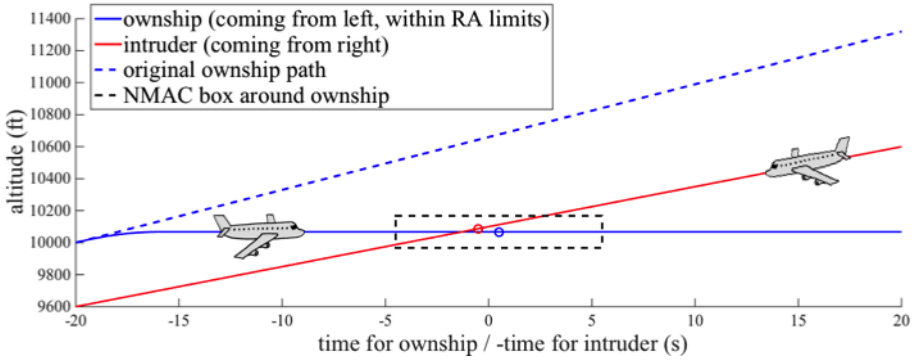

Fig. 7. Original ownship path (cyan) and intruder path (red) vs. ownship responding to a do-not-climb (DNC) advisory issued by the ACAS X tables in starting state: $r = 4{,}000$ ft, $r_v = 200$ ft/s, $\theta_v = 180°$, $h = 600$ ft, $\dot{h}_0 = 1{,}980$ ft/min, $\dot{h}_1 = -1{,}500$ ft/min.

regions stated in Fig. 3. Overall, $4{,}461{,}600$ discrete states were examined, among which $44{,}306$ states (1.2%) did not meet the conditions of Fig. 3: $11{,}524$ of these were unresolvable, i.e., the intruder was too close for any advisory to avoid NMAC; while $32{,}782$ could have been resolved with a different safe advisory that satisfies Theorem 1.

Our analysis led to the identification of unexpected behavior in the ACAS X lookup tables. In some cases, the ACAS X advisory seems to *induce* an NMAC (Fig. 7), i.e., if the initial advisory is not strengthened or reverted later, an NMAC will occur. In other cases, the advisory does not seem to have any benefit, that is flying at vertical rates disallowed by the advisory would actually avoid NMAC while not all allowed vertical rates are safe. Of course, such unsafe advisories would be disallowed by our safe regions. Notice that these behaviors are not necessarily all deemed undesirable, as ACAS X tries to minimize alerting the pilot unless it has to do so; for some cases, ACAS X will strengthen the advisory later and hence does not issue a disruptive alert immediately. Fig. 7 depicts a typical example where the ACAS X advisory induces an NMAC. The ownship is flying from the left and the intruder from the right. As time counts down, the intruder evolves towards the ownship and an NMAC happens at $t = 0$. The original path of the ownship does not lead to an NMAC. However, ACAS X gives a Do-Not-Climb advisory. If the pilot, following this advisory, decides to stop climbing, its trajectory will cause an NMAC. (Other examples are in Technical Report [10].)

The development of the safe regions gave an insight into possible improvements for the ACAS X system. Although we are not analyzing the complete system, nor the subsequent advisories, we automatically pointed out some subregions of the state space worth looking at. Some of those problems were independently identified by the ACAS X team using simulation-based testing, and will be addressed in subsequent revisions of the system. When extended to check contiguous regions of the state space, our approach will have the potential for a complete analysis of the system over all potential encounter configurations, thereby reducing vulnerability to the sampling of encounter scenarios.

7 Related Work

Kochenderfer and Chryssanthacopoulos [12] describe the design of the ACAS X lookup-tables. Their principled approach, based on optimizing an MDP, guarantees the selec-

tion of optimal advisories according to a cost model. The state space and dynamics are discretized. Their notion of optimality depends on costs assigned to various events.

Von Essen and Giannakopoulou [3] use probabilistic model-checking to analyze an MDP based on [12]. They investigate the probability of several undesirable events occurring. Because they ostensibly analyze an MDP, their work inherits many of the assumptions of ACAS X, including discretized dynamics. Their analysis depends heavily on the MDP considered and thus needs to be redone on every version of ACAS X.

Lygeros and Lynch [16] use hybrid techniques to formally verify the TCAS conflict resolution algorithms. They assume—rather than prove—that TCAS ends up in a state where one aircraft has a climbing advisory and the other a descending advisory. They then prove (by hand) a lower bound on the vertical separation of both aircraft at the point of closest approach. In contrast, we do not assume anything on ACAS X's advisories.

Holland *et al.* [9] and Chludzinski [1] simulate large numbers of encounters, including tracks from recorded flight data, to evaluate the performance of ACAS X. These simulations account for high-fidelity details of an encounter, but they only cover a finite set of the continuous state space with no formal guarantees.

Tomlin *et al.* [22], Platzer and Clarke [20], Loos *et al.* [15] and more recently Ghorbal *et al.* [8] use hybrid systems approaches to design safe horizontal maneuvers for collision avoidance. Dowek *et al.* [2] and Galdino *et al.* [7] describe and verify in the PVS theorem prover a collision avoidance system of their design called KB3D.

Overall, our approach is different from previous complementary work in that:

- unlike [3,12], we rely on an independent model from the one used to design ACAS X;
- unlike [2,7,8,15,20,22] we analyze an independent industrial system and not a safe-by-design system;
- unlike [2,3,7] our analysis uses realistic, continuous dynamics;
- unlike [16,22] we provide universal safe regions that can be reused for new versions of ACAS X or even for new systems;
- unlike [1,9,11,16,22], we provide mechanized proofs of correctness of our model.

8 Conclusion and Future Work

We developed a general strategy for analyzing the safety of complicated, real-world collision avoidance systems, and applied it to ACAS X. Our strategy identifies safe regions where an advisory is proved to always keep the aircraft clear of NMAC, under some assumptions. We identified states where ACAS X is provably safe, and fed others showing unexpected behaviors back to the ACAS X development team. The identified safe regions are independent from the version of ACAS X and can thus be reused for future versions. In future work, we plan to extend our hybrid model to account for curved trajectories of both aircraft as well as vertical acceleration of the intruder.

Acknowledgments. The authors would like to warmly thank Stefan Mitsch and Jan-David Quesel for their support of the KeYmaera tool. The authors would also like to thank Jeff Brush, Jessica Holland, Robert Klaus, Barbara Kobzik-Juul, Mykel Kochenderfer, Ted Londner, Sarah Loos, Ed Morehouse, Wes Olson, Michael Owen, Joshua Silbermann, Neal Suchy, and the ACAS X development team for interesting remarks.

References

1. Chludzinski, B.J.: Evaluation of TCAS II version 7.1 using the FAA fast-time encounter generator model. Tech. Rep. ATC-346, MIT Lincoln Laboratory (April 2009)
2. Dowek, G., Muñoz, C., Carreño, V.: Provably safe coordinated strategy for distributed conflict resolution. In: AIAA Guidance Navigation, and Control Conference and Exhibit (2005)
3. von Essen, C., Giannakopoulou, D.: Analyzing the next generation airborne collision avoidance system. In: TACAS, LNCS, vol. 8413, pp. 620–635. Springer (2014)
4. Federal Aviation Administration: Introduction to TCAS II (February 2011), version 7.1
5. Federal Aviation Administration TCAS Program Office: Algorithm design description for the surveillance and tracking module of ACAS X (July 2014), run12
6. Federal Aviation Administration TCAS Program Office: Algorithm design description for the threat resolution module of ACAS X (May 2014), version 3 Rev. 1
7. Galdino, A., Muñoz, C., Ayala, M.: Formal verification of an optimal air traffic conflict resolution and recovery algorithm. In: WoLLIC. LNCS, vol. 4576. Springer (2007)
8. Ghorbal, K., Jeannin, J.B., Zawadzki, E., Platzer, A., Gordon, G.J., Capell, P.: Hybrid theorem proving of aerospace systems: Applications and challenges. Journal of Aerospace Information Systems (2014)
9. Holland, J.E., Kochenderfer, M.J., Olson, W.A.: Optimizing the next generation collision avoidance system for safe, suitable, and acceptable operational performance. Air Traffic Control Quarterly (2014)
10. Jeannin, J.B., Ghorbal, K., Kouskoulas, Y., Garnder, R., Schmidt, A., Zawadzki, E., Platzer, A.: A formally verified hybrid system for the next-generation airborne collision avoidance system. Tech. Rep. CMU-CS-14-138, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA (2014), `http://reports-archive.adm.cs.cmu.edu/anon/2014/CMU-CS-14-138.pdf`, KeYmaera files available at `http://www.ls.cs.cmu.edu/pub/acasx.zip`
11. Kochenderfer, M.J., Espindle, L.P., Kuchar, J.K., Griffith, J.D.: Correlated encounter model for cooperative aircraft in the national airspace system version 1.0. Tech. Rep. ATC-344, MIT Lincoln Laboratory (October 2008)
12. Kochenderfer, M.J., Chryssanthacopoulos, J.P.: Robust airborne collision avoidance through dynamic programming. Tech. Rep. ATC-371, MIT Lincoln Laboratory (January 2010)
13. Kochenderfer, M.J., Holland, J.E., Chryssanthacopoulos, J.P.: Next generation airborne collision avoidance system. Lincoln Laboratory Journal 19(1), 17–33 (2012)
14. Kochenderfer, M.J., Monath, N.: Compression of optimal value functions for Markov decision processes. In: Data Compression Conference. Snowbird, Utah (2013)
15. Loos, S.M., Renshaw, D.W., Platzer, A.: Formal verification of distributed aircraft controllers. In: HSCC. pp. 125–130. ACM (2013)
16. Lygeros, J., Lynch, N.: On the formal verification of the TCAS conflict resolution algorithms. In: IEEE Decision and Control. vol. 2, pp. 1829–1834. IEEE (1997)
17. Platzer, A.: Differential dynamic logic for hybrid systems. J. Autom. Reas. 41(2), 143–189 (2008)
18. Platzer, A.: Logical Analysis of Hybrid Systems: Proving Theorems for Complex Dynamics. Springer (2010)
19. Platzer, A.: Logics of dynamical systems. In: LICS. pp. 13–24. IEEE (2012)
20. Platzer, A., Clarke, E.M.: Formal verification of curved flight collision avoidance maneuvers: A case study. In: FM. LNCS, vol. 5850, pp. 547–562. Springer (2009)
21. Platzer, A., Quesel, J.D.: KeYmaera: A hybrid theorem prover for hybrid systems. In: IJCAR. LNCS, vol. 5195, pp. 171–178. Springer (2008)
22. Tomlin, C., Pappas, G.J., Sastry, S.: Conflict resolution for air traffic management: A study in multiagent hybrid systems. IEEE Transactions on Automatic Control 43(4), 509–521 (1998)